Proceedings of ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2014
August 17-20, 2014, Buffalo, New York, USA

DETC2014-34390

TOPOLOGY PRESERVING DIGITIZATION OF PHYSICAL PROTOTYPES USING DEFORMABLE SUBDIVISION MODELS

Erhan Batuhan Arisoy
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: earisoy@andrew.cmu.edu

Levent Burak Kara
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: lkara@cmu.edu

ABSTRACT

Physical prototyping is an important stage of product design where designers have a chance to physically evaluate and alter digitally created surfaces. In these scenarios, designers generate a digital model, manufacture and alter the prototype as needed, and redigitize the prototype through scanning. Despite the variety of reverse engineering tools, redigitizing the prototypes into forms amenable to further digital editing remains a challenge. This is because current digitization methods cannot take advantage of the key elements of the original digital model such as the wireframe topology and surface flows. This paper presents a new reverse engineering method that augments conventional digitization with the knowledge of the original digital model's curve topology to enhance iterative shape design activities. Our algorithm takes as input a curve network topology forming a subdivision control cage and a 3D scan of the physically modified prototype. To facilitate the digital capture of the physical modifications, our algorithm performs a series of registration, correspondence and deformation calculations to compute the new configuration of the initial control cage. The key advantage of the proposed technique is the preservation of the edge flows and initial topology while transferring surface modifications from prototypes. Our studies show that the proposed technique can be particularly useful for bridging the gap between physical and digital modeling in the early stages of product design.

Introduction

Rapid exploration of new and aesthetic shape ideas has been an indispensable part of current product design process. Nearly seventy percent of final product forms and life cycle cost have been determined during this iterative activity [1]. Despite widespread use of digital modeling and editing tools, many artists still find the time spent for physical externalization of design ideas, known as prototyping, invaluable and creativity stimulating. It can not be fully eliminated throughout modeling sessions. For example, a car designer summarizes the importance of prototyping as follows: " You drag your hand on a particular clay model section, surface (*e.g.* side body or front hood) gives you the feeling of the form evolution, it gives you the drive of shaping that volume to give it grace and elegance or sculpted concave effects for a more nervous signature [2]." In a general scenario, designers generate digital models, edit them, fabricate prototypes and modify, and redigitize for further digital modeling. According to our discussions and observations with the artists at Carnegie Mellon University's Design Studio, modifications that can be done on physical prototypes can be classified into four main categories: (1) material removal through sanding or machining operations (like sculpting), (2) component addition (buttons, keys), (3) genus changes (holes) and (4) feature line introduction (crease lines). Figure 1 illustrates some example prototypes.

Although, existing CAD and CAE tools provide rapid conversion from physical to digital and digital to physical, these both way conversions are not only costly but they also require repet-

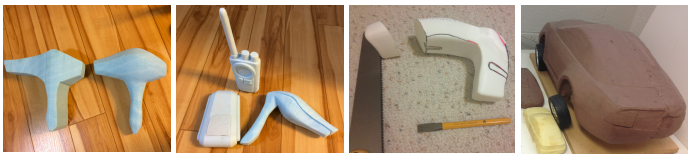


FIGURE 1. Modification examples on prototypes: a) smoothing, b) feature and component addition, c) material removal and d) genus change.

itive and labor intensive tasks [3, 4]. While many researchers have recognized novelty of coupling digital and physical modeling sessions, proposed reverse engineering tools generate models oblivious to user defined surface flows (chain of consecutive edges) and wireframe topology. In other words, these tools construct a new surface topology on the reconstructed model that is usually more complex than the original and violates initial surface flows. To overcome this shortcoming, we present a new reverse engineering method that takes as input a curve network topology forming a subdivision control cage S_1 and a 3D scan S_2 of physically modified prototype. Our algorithm performs a set of registration, correspondence calculation and deformation operations to calculate a new configuration of the initial control cage S_1. This control cage is utilized for approximate subdivision surface generation [5] which exhibits deformations done on physical mock-ups. Key advantages of the proposed technique are preservation of initial edge flows and wireframe topology and formulation of this reverse engineering problem in a linear scheme while transferring modifications on physical prototypes. In summary, proposed method has following four contributions:

1. A wireframe assisted subdivision surface deformation algorithm
2. Preservation of wireframe topology and initial edge flows while digitizing physical prototypes
3. A new evaluation metric to measure amount of geometric deformations applied in physical modeling session relative to the initial digital design
4. Providing effortless transitions between physical and digital media and bridging the gap between different design streams

Related Work

In this section, we categorize previous studies in three different groups: (1) rigid and non-rigid body registration techniques, (2) geometric deformation algorithms and (3) reverse engineering tools.

Rigid and Non-rigid Body Registration: A large number of studies have been devoted to rigid body registration problem since early 1970's. Especially with widespread use of 3D scanners, these algorithms are utilized for shape completion and tracking from incomplete and noisy 3D data. The key assumption in these studies is that relative transformations between two 3D models can be described by a single Euclidean transformation θ. The most well-known algorithm for rigid body alignment is Iterative Closest Point (ICP) proposed by Besl and McKay [6]. This algorithm initializes a mapping by finding closest point on both models and iteratively calculates a rigid body transformation to register these two models by minimizing least square error. Techniques to improve convergence of rigid ICP algorithms are summarized in [7]. Gelfand *et al.* [8] improved alignment performance and final quality by selecting feature points that constrain all degrees of freedom in the rigid-body transform [8–12]. Unfortunately, these methods do not guarantee that final mapping is smooth and bijective (two different points on the same surface can be assigned to the same point on the second surface), and they require a good initial alignment to succeed in most cases. Thus, they are more suitable to pairwise registrations with multiple 3D scans. Allen *et al.* [13] used an affine transformation at every vertex of the source model to allow non-rigid registration of full-body scans to high-resolution templates. Chui and Rangarajan [14] iteratively compute pointwise correspondences using a softassign framework and thin-plate splines.

Geometric Deformations: Earlier studies in geometric deformation field focused on utilization of space deformations that can handle arbitrary surface representations. Bechmann [15] provides a survey of such techniques. The main motivation is embedding input geometry in a discrete space and defining a wrapping field to deform the underlying model in this embedded space. However, modifications that can be obtained through these deformation schemes are too limited, difficult to control and unsuitable to data fitting. In order to alleviate these shortcomings, new deformation methods capable of preserving intrinsic properties are proposed. These techniques can be classified according to the deformation models they employ such as isometric models [11, 16], as rigid as possible models [17] and skeleton driven models [18]. Most prominent techniques include but not limited to physically inspired deformation models and differential coordinates (Laplace and gradient-based representations) [19] that are considered as linear deformation techniques. These schemes may fail at capturing non-rigid deformations because of their linear formulations. As a step towards solving this problem, a non-linear alternative *PriMo* is introduced by Botsch and colleagues [20]. Sorkine and Alexa [21] demonstrated that local rigidity can be achieved by enforcing non-translational components to maximize their rotations. Similarly, embedded deformations [22] alleviated dependency on geometrical representations by generating a temporary graph structure on input 3D models.

Reverse Engineering Tools: A multitude of computational tools are devoted to automatic generation of polygonal, implicit or parametric surfaces from set of 3D point scans [23–25]. Lavou *et al.* [26] proposed a technique to fit subdivision surface con-

trol cages to polygonal meshes. Similarly, Dammertz *et al.* [27] presented an adaptive surface construction method with a hierarchy of quadrilateral meshes adapting surface topology. However, these algorithms result in a 3D surface with arbitrary topology. With recent advances in animation industry, designers aimed to generate specific surface flows on reconstructed surfaces that are suitable to subsequent deformations for animation [3, 28]. Takayama [29] presented an interactive system to create user-defined surface flows consisting of quad meshes by enabling designers to easily sketch patch boundaries on digital models. These quad patches serve as a control cage and the system generates a subdivision surface later on. Although these tools support retopologization of reconstructed models using sketch-based interfaces, this operation is still laborious and challenging for novice users. Our distinguishing difference from existing reverse engineering and surface deformation tools is preservation of initial user-defined topology and edge loops while digitizing modifications on physical prototypes. Figure 2 demonstrates an example of user drawn 3D curve network (a), corresponding subdivision surface (b) and extracted topology from this surface if we use Geomagic (c) . It is clear that the new topology identified by Geomagic is very complicated compared to the original, as it lacks topology information about initial surface flows.

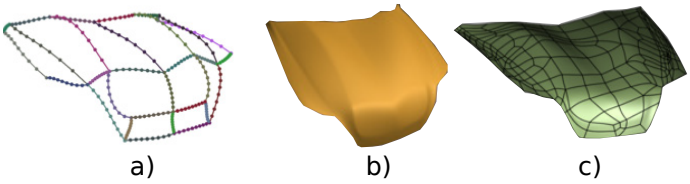


FIGURE 2. Comparison between user-defined curve network and topology obtained by Geomagic: a) User-defined topology, b) subdivision model and c) topology obtained by Geomagic.

Overview and Methodology

In a typical scenario, our algorithm takes as input a 3D curve network forming a subdivision control cage (source model) S_1 and a 3D scan representing physical prototype in digital world. Our methodology consists of three main blocks: registration, correspondence calculation and surface deformation. Initially, our registration step enables us to approximately align the control cage S_1 with the polygonal model S_2 extracted from 3D scans. To do so, we use a modified version of the ICP algorithm [6] via user assistance, where user paints corresponding two regions on S_1 and S_2 that are not experienced any updates in physical modeling. In the next step, a pairwise correspondence mapping is calculated between these two digital models based on the following assumption: deformations done on the physical prototypes are approximately isometric (the geodesic distance between any two points on both surfaces are approximately same). We utilized geodesic distance based matching algorithm presented by Huang *et al.* [30]. In the final step, surface deformation formulates this reverse engineering problem in a linear least square scheme. Our algorithm considers initial control grid as a deformable subdivision surface and drives this deformable model. New control point configurations are expected to form a subdivision surface that matches the target polygonal model. This deformation scheme employs not only point to point correspondences but also initial topology information to regularize vertex distribution on the entire surface. Figure 3 illustrates overall pipeline of the proposed algorithm.

Generation of Initial Control Cage (Source) and Polygonal Model of the Prototypes (Target)

Our work is stimulated by the observation that a product design form can be described using a sparse collection of 3D curves. Surface dictated by this initial curve network can be considered as a proxy to evaluate, modify and explore different design candidates. In our approach, we employed the interface developed by Orbay *et al.* [31] to generate an initial network topology using epipolar sketching scheme where each curve is represented as cubic Bezier curves. This topology defines initial edge flows and forms a control cage for approximate Catmull-Clark subdivision surfacing [32]. Afterward, the algorithm automatically identifies closed curve loops and constructs Coons patches surfacing them. Transition from these Coons patches to subdivision surface control cage is realized through vertex sampling on these parametric patches with user defined resolution. Then, user can easily output an STL model and generate physical prototype either using a CNC machine, 3D printer or rapid prototyping machine. After generation of physical prototypes, artists can perform desired physical operations such as sanding, cutting, sculpting or drilling. In proposed pipeline, we obtain 3D scans of these prototypes using a NextEngine 3D laser scanner. We performed all scans with a resolution of eighty points per inch and from twelve different view points to generate complete 3D models. The built-in scanner software is utilized to repair and mesh 3D point sets.

Geodesic Distance Distribution (GDD) Based Deformation Metric

In order to measure amount of deformations a digital model undergone in the physical modeling session, we propose a new surface deformation evaluation metric based on the existing work from [33]. Basic question we are trying to answer is: how much initial geometric model is changed? The evaluation metric to answer this question should satisfy following three requirements: (1) scale invariance, (2) orientation invariance, (3) resolution and topology invariance. These requirements are necessary since initial control cage is both geometrically and topologically indifferent than the 3D scan of the modified prototype. All three requirements can be satisfied using a geodesic distance-based continuous function $\mu(v)$ proposed by Hilaga *et al.* [33]. This function

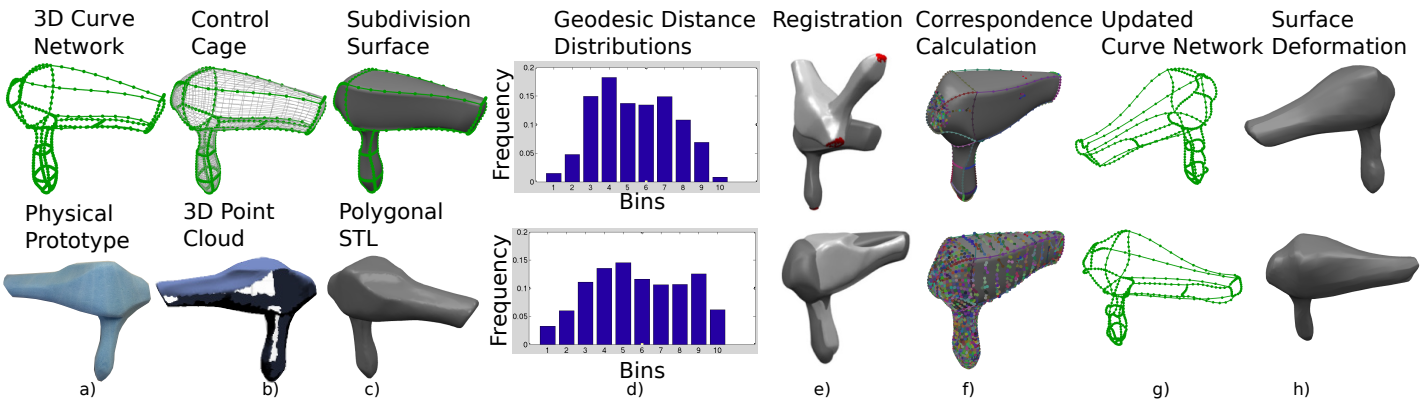


FIGURE 3. A breakdown of our approach: a) user-drawn 3D curve network (top) and physical prototype, b) quadrangulated face loops for subdivision cage (top) and 3D point cloud, c) initial subdivision surface (top) and polygonal model extracted from 3D scan d)geodesic distance frequency histograms for subdivison surface (top) and polygonal stl, e) registration, f) correspondence calculation, g) updated curve network and h) deformation result.

is utilized for topology extraction using Reeb graphs and shape matching. Main idea is resampling a user defined number of n seed points $B = \{b_0, b_1, b_2 ..... b_n\}$ on both surfaces and calculating the sum of shortest geodesic distances from these seed points to every other seed point. These distances are weighted by the area of their one ring neighborhood. The discretized version of $\mu(v)$ is calculated using following Equation 1 for every vertex $v \in S$.

$$\mu(v) = \sum_{\substack{j=1 \\ b_i \in B, a \in S}}^{n} d(v, b_i) * area(b_i) \qquad (1)$$

After normalization for scale invariance, larger function values for a vertex indicate that the point is close to geodesic surface center of the input model. In order to measure the deformation difference between two digital models, we extended this formulation by calculating histograms to generate a geodesic distance distribution signature for each individual model. To do so, we generate histograms with fixed number of bins and divide each bin count by the total number of sampled points to have resolution independence. We believe that variation between histograms of two different geometric models will give us a metric to measure their diverse geodesic distance distributions. Most common way of measuring histogram differences is achieved using Earth Mover's Distance (EMD) metric [34]. EMD quantifies minimal amount of work that must be completed to change one histogram into the other by moving its distribution mass. Equation 2 shows our final metric and Figure 4 demonstrates histograms that are calculated from two example models where the calculated EMD is 22.39.

$$GDD(S_1, S_2) = d_{EMD}(hist(\mu_n(S_1)), hist(\mu_n(S_2))) \qquad (2)$$

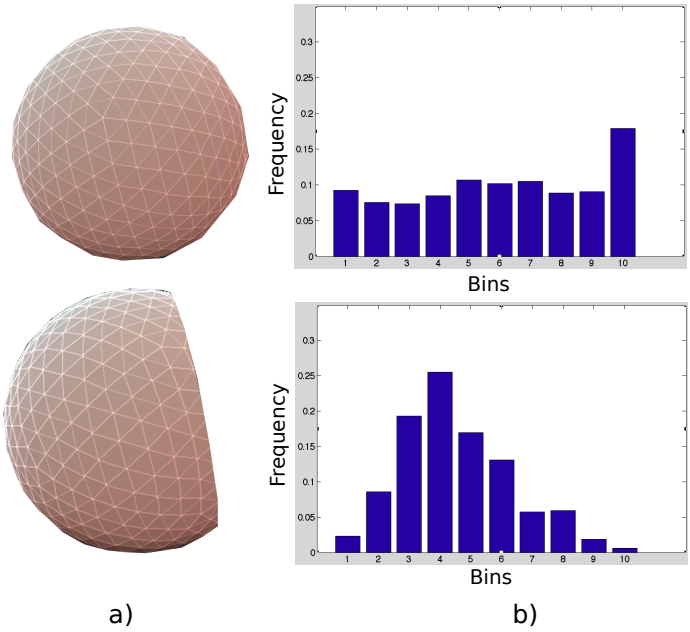


FIGURE 4. Geodesic Distance Distribution Examples: a) input geometries, b) frequency histograms.

Initial Registration

Polygonal models extracted from 3D point scans need to be aligned with initial control cages in order to obtain accurate correspondence calculations in subsequent steps. Initially, we utilized ICP algorithm developed by Besl and McKay [6] that is guaranteed to converge. For each vertex $v_i \in S_1$, ICP algorithm calculates the closest point k_i on the target surface S_2. Then, the algorithm returns the optimal rigid motion θ which minimizes the distance between $\theta(v_i)$ and k_i. Instead of minimizing pointwise distances, it is more reliable to minimize distances between

points to planes, especially if there are large flat regions on both models S_1 and S_2. We provide a painting tool where user can mark two corresponding regions on both models and Figure 5 demonstrates some alignment examples where red regions are user painted areas to initialize registration.

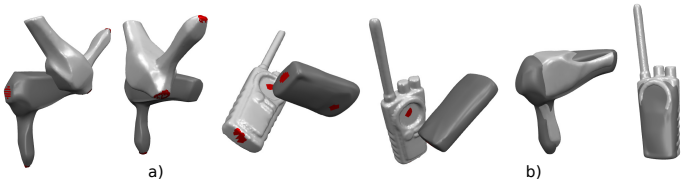


FIGURE 5. User assisted registration examples: a) initial configurations, b) registration results.

Correspondence Calculation

A reliable correspondence calculation is crucial for success of a subsequent deformation algorithm. Pairwise correspondences define how initial source control cage S_1 has to be updated in order to match the target model S_2. In this section, S_1 represents control cage of the initial digital SubD model and S_2 stands for the polygonal surface extracted from 3D scans of physical mock-up. Our correspondence calculation algorithm assumes that initial model is undergone approximate isometric deformations. Therefore, geodesic distances between vertices on S_1 should be similar to distances between their correspondences on S_2. In other words, if two points v_i and v_j on S_1 have a geodesic distance $d(v_i, v_j)$, then their corresponding points found on S_2, k_i and k_j, are expected to have a $d(k_i, k_j)$ value closer to $d(v_i, v_j)$. Overall pipeline of correspondence calculation algorithm is summarized in Figure 6.

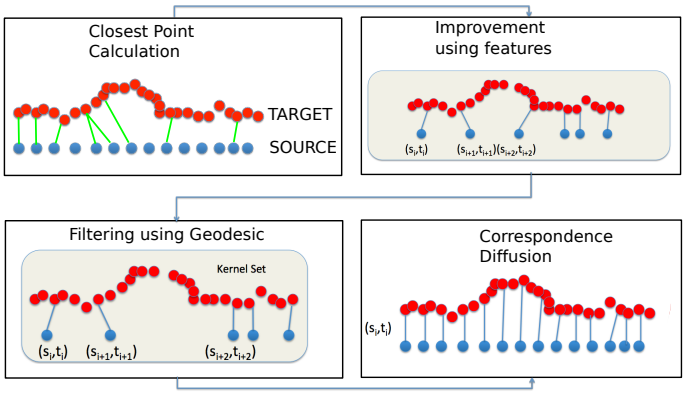


FIGURE 6. Entire correspondence calculation demonstration in 2D.

Geodesic distances on both surfaces are approximated as graph distances and calculated using Dijkstra's shortest path algorithm. Our correspondence calculation algorithm is based on the method proposed by Huang *et al.* [30] for non-rigid deformation of partially overlapping 3D surfaces. In the first step, initial correspondences are obtained by creating a set of points that are close to each other not only in Euclidean distance space but also in a geometric feature space. In the second step, these initial correspondences are filtered out using a spectral method [35], which eliminates correspondences violating the geodesic distance consistency. Next step is diffusing these correspondences to the rest of the model similar to a heat diffusion problem. Detailed information can be obtained in the reference paper.

Surface Deformation

Although our deformation scenario explained below is based on utilization of approximate subdivision surfaces, because of their smoothness and linear properties, overall pipeline is applicable to any digital surface representation. Approximate Catmull-Clark subdivision surfaces, representation that we use, are actually generalization of bicubic B-spline surfaces to arbitrary topologies. Thus, they share similar properties such as smoothness and parametric properties. Main advantage of this approximate representation is that limit positions of control cage vertices can be calculated via linear weight masks presented in [5]. Thanks to linear nature of these masks, final limit positions of subdivision surfaces can be formulated using a linear system of equations where an assembly matrix encodes all neighborhood voting scheme. Final limit positions are defined as a linear combination of the columns of this assembly matrix with control cage vertices. In order to solve the deformation problem, we seek to obtain unknown new control vertex positions that result in limit surface point configurations close to the 3D scan of prototypes. Equation 3 represents linear least square formulation.

$$E_{obj} = \sum_{v_i \in S_1}^{n} ||Ax - b||^2 \tag{3}$$

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MM} \end{bmatrix}}_{\text{Limit Position Mask}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}}_{\text{Control Vertices}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}}_{\text{Correspondence Positions}}$$

After calculation of correspondences for every vertex v_i on surface S_1, we minimize energy functional in Equation 3 to calculate new configuration of control vertices. v_i are vertices on the undeformed subdivision surface S_1 and matrix A is the assembly matrix with size $M \times M$. x stands for new position of control vertices ($M \times 1$) and b represents target positions where we want limit positions to be after deformations ($M \times 1$).

This optimization problem can be solved using normal equations where new positions of control vertices are given by

$(A^TA)^{-1}A^Tb$. Control vertices of the initial SubD mesh are iteratively deformed until they attain a stable configuration. That means our limit surface approximates the target surface closely. In each iteration, control vertices experience not only external forces coming from correspondences but also internal Laplacian forces that aim to preserve smoothness of the control polygon during this iterative process.

Importance of Topology Information: User drawn curve loops constituting wireframe topology play a key role to convey the intended design with minimum amount of information. It also enables us to regenerate entire SubD surface using new configurations of these curves. Topological regularization is crucial, especially if large amount of material is removed or added in prototyping sessions. For example, in order to replicate cut operation, our deformable model shrinks on itself which results in control vertices sitting on top of each other. Having the topology beforehand enables us to track new positions of user drawn curves after the deformation and the system can regenerate the underlying subdivision surfaces with new control cage configurations. This way, we are able to replicate not only material removal operations but also material additions without topological updates.

Deformation rule: Our deformation approach is summarized in Figure 7. We start with an initial set of correspondences and then iteratively calculate new positions of control vertices. Because of possible errors in correspondence calculation algorithm, we do not directly use these positions. Instead, we combine them with Laplacian smoothing in order to smooth out possible matching errors. Complete pipeline of the proposed algorithm can be summarized in the following pseudocode.

Algorithm 0.1: SURFACEDEFORMATION$(S_1, S_2, tolerance)$

$C = \{c_0, c_1, ..., c_n\}$
Initial curve network where each c_i is a Cubic Bezier
$(S_1', S_2') \leftarrow Registeration(S_1, S_2)$
$R = \{(v_0,k_0),(v_1,k_1).....,(v_m,k_m)\} \leftarrow InitCorrespondence(S_1', S_2')$
R is Initial High Confidence Matches
$P = \{(v_0,k_0),(v_1,k_1).....,(v_m,k_m)\} \leftarrow DiffuseCorrespondence(L)$
P is Extended Matches
$iteration \leftarrow 1$
while $tolerance > 0.05$

do
- minimize $E_{obj} = \sum_{\vec{v}_i \in S_1'}^{n} ||Ax - b||^2$
- $v_i(\vec{t+1}) \in S_1' \leftarrow (A^TA)^{-1}A^Tb$
- $\triangle \vec{v}_i^{\,Deform} = \vec{v}_i(t) + \alpha_1 \triangle \vec{v}_i(t)^{\,Laplace} + \alpha_2 \vec{v}_i(t+1)$
- $\vec{v}_i(t+1) \leftarrow \triangle \vec{v}_i^{\,Deform}$
- **if** mod(iteration,10)=0
  - **then** $RegenerateSubd(v_i(\vec{t+1}))$
- $iteration \leftarrow iteration + 1$

return $(\vec{v}_i(t))$

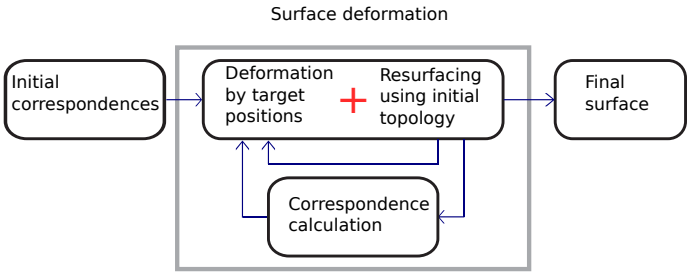


FIGURE 7. Breakdown of our deformation approach.

Results and Discussions

Figure 8 demonstrates our test results on a regular dodecahedron structure. Initially, we created a dodecahedron with an edge length of 1 unit with 3K vertices. Then, we manually introduced several free form deformations on this dodecahedron. Displacement maps show the amount of deformation applied to the regular dodecahedron. Red color represents large displacements. Blue colored regions did not experience any deformation relative to the regular dodecahedron model. Figure 8 demonstrates the results where for all three cases, our error was below 1% of the diagonal length of the bounding boxes of each model and the Earth Mover's Distance between the geodesic distance distribution histograms are below 1 unit (0 means identical geodesic distance distribution).

For a more systematic evaluation, we implemented the free form cage deformation algorithm (FFD) [36] to generate an artificial set of deformed dodecahedron models using random parameters for the cage vertex displacements. In all of our results, after the deformations, the Earth Mover's Distance between geodesic distance histograms are below 0.5, and our algorithm was able to successfully transfer modifications introduced on the original models. Table 1 summarizes details of our synthetic test results and the results are consistent with our expectations that the accuracy decreases as the amount of deformation increases.

# of Models	# of Vertices	Deformation Amount	Accuracy
25	4200	1%	0.001%
25	5000	3%	0.012%
25	4600	7%	0.156%
25	4200	10%	4%

TABLE 1. Test results.

Similarly, Figure 9 illustrates two example models generated through FFD deformations and the deformed versions of these artificially generated geometries. In the first row, we demonstrate

the algorithm on a car hood design. The starting point is a curve network conveying a flat hood design and the control cage obtained from this curve network. We assume that in the physical world this hood is inflated around the center region of the model. This can be seen in the forth column where this deformation is generated through the FFD algorithm. In the second example in Figure 9, we illustrate the virtual-physical design of a computer mouse. We demonstrate the effect of material addition on one side of the model for ergonomic purposes. Once critical observation in these results is the fact that the resulting deformed digital models have surface flows similar to the original ones. In addition, updated curve network topologies support this observation and as such the only subset of the curve networks are reconfigured to transfer modifications done in the FFD phase.

In addition to these examples, a student designer from CMU Design Studio used our tool for the design of a mouse with a roller button. Figure 11 demonstrates this case study. First, the designer starts with an initial curve network topology, and generates digital SubD models using existing CAD tools. Then, she fabricates a physical prototype of this initial model and applies modifications such as sanding for smoothing and shape modification. In Figure 11.e, 3D scan of the new prototype is shown. The major modifications are applied to both sides of prototype 1. Last row in Figure 11 presents the resulting curve network with the corresponding digital SubD after the use of the proposed methods. We are currently conducting further studies to assess the efficacy of the proposed method.

Implementation Details and Limitations

Figure 10 illustrates real product form design examples where each example is created under 3 minutes on a 2GHz machine. In order to perform search queries efficiently, we employed a KD tree structure (NlogN). In addition, the bottle neck of our computations is the correspondence finding. The shortest distances between two vertices are assumed to be graph distances and are calculated using Dijkstra's shortest path algorithm. The iterative surface deformation algorithm is terminated when the geodesic distance distribution difference between two subsequent iterations is below 0.1.

Clearly, the resulting digital surfaces *approximate* the scanned physical surfaces. However, this is useful to suppress undesirable noise arising from either 3D scanning devices or the physical prototypes. This approximating behavior can be seen in all three results in Figure 9. Another critical consideration is the location of crease curves after the deformation iterations. Although the resulting surfaces can represent the alterations done on the prototypes with sufficient accuracy, the exact locations of the crease curves might deviate from their physical counterparts because of the iterative nature of our algorithm. Similarly, our algorithm does not preserve some of the geometrical properties such as curvature continuity or monotonic curvature that might

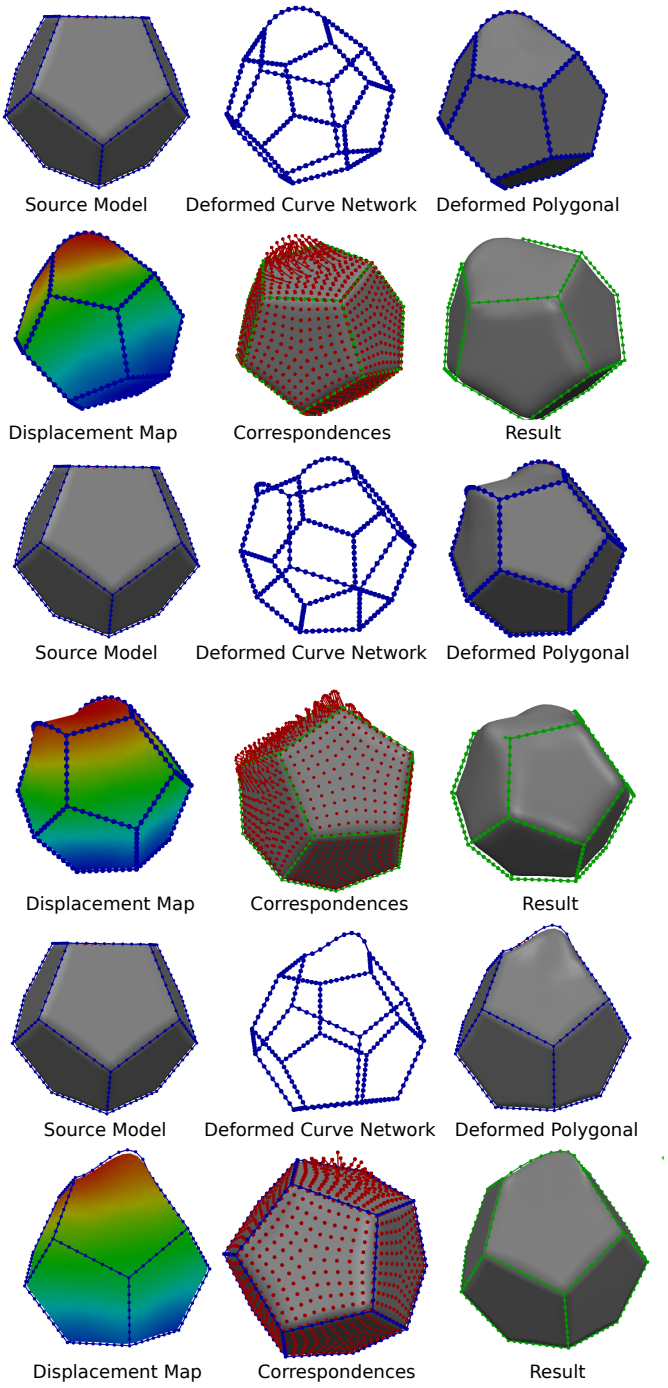


FIGURE 8. Testing on a regular dodecahedron.

be important for the designers. This challenge can be resolved by post processing operations such as V-spring smoothing. The main advantage of the proposed work is the preservation of the edge loops and initial topology in subsequent steps of the deformation. This facilitates not only regularization but also different

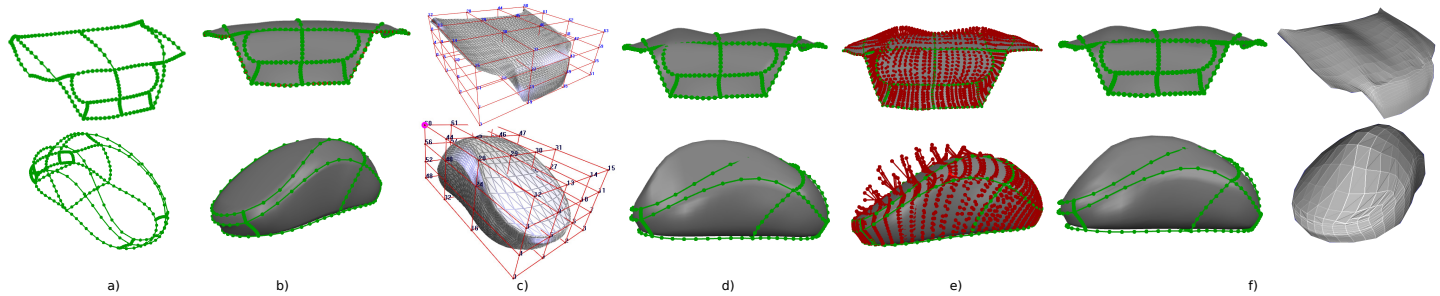


FIGURE 9. Car hood and mouse example through FFD deformation: a) initial curve topology, b) SubD model, c) FFD cage, d) FFD deformed SubD model, e) correspondences and f) deformation results.

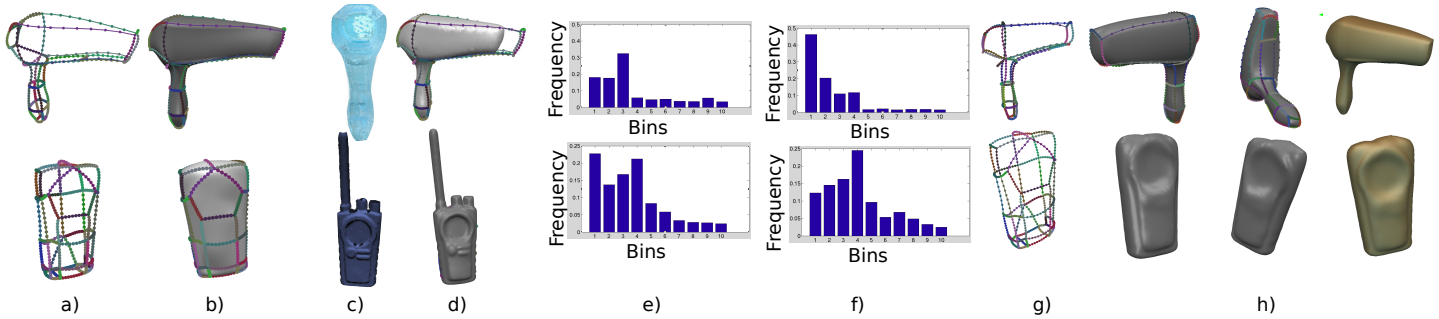
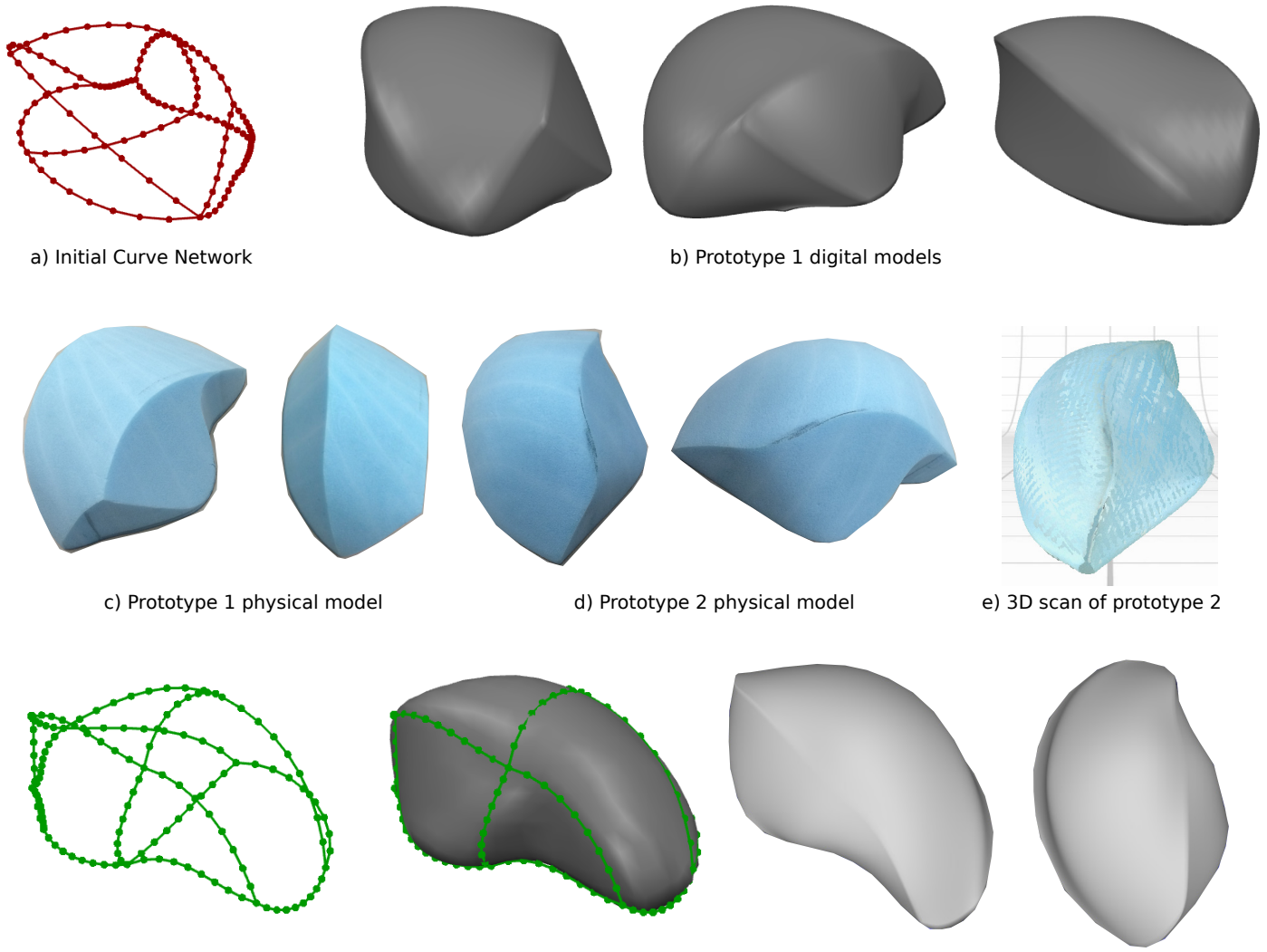


FIGURE 10. Different example cases for product form design: a) initial wireframe topology, b) initial SubD model, c) 3D scans, d) polygonal model extracted from 3D scans, e-f) geodesic distance distribution histograms of the initial model and the physical prototype, g) updated topology and h) resulting deformed models.

modifications performed in the physical world. In the first material removal example, the new curve network is updated so that it reflects a cut operation. At the same time, regions that are not affected in the physical modeling session such as the handle, did not experience any topology change. On the other hand, on the second row, the modifications done on the walkie-talkie model did not require any topological updates. The deformations were best represented by only repositioning the vertices of the control cage. In this scenario, there is no need to update the initial network flow and our deformation algorithm behaved the way we expected. The positions of the initial curve vertices are not changed, only the positions of the inner loop vertices are updated according to the 3D scans. These are all desirable properties thanks to the utilization of the initial curve network topology. However, this utilization might be a limitation in certain cases. For example, if the designer introduces characteristic features such as crease curves onto the mock-up model, then topology addition is required in order to be able to geometrically represent these new feature lines on the final surface.

Conclusions and Future Work

We present a new method that allows shape modifications performed in the physical world to be captured digitally in ways that facilitates the digital workflow. Specifically, we showed how a digitally created concept model could be physically realized and reverse engineered back into the digital space, while remaining congruent with the original digital model. Our results indicate that the proposed algorithm is effective for physical alterations that do not induce a topology change. However, the nature of physical modeling involves a wide variety of different modifications like genus changes. In order to accommodate different types of transformations, our initial network topology should detect these differences to introduce such changes automatically. Our studies have also indicated another challenge: most artists do not use consistent scales for prototyping. In order to accommodate these scale differences, our algorithm currently normalizes (via uniform scaling) input object into a unit sphere. However, this scaling might not work especially for large material removals from the initial concept. In future work, we plan to improve our surface deformation algorithm so that the initial topology will be malleable. This curve network should be able to update and adjust itself according to the detected modifications on the physical prototype. Additionally, we plan to perform user studies to



a) Initial Curve Network

b) Prototype 1 digital models

c) Prototype 1 physical model

d) Prototype 2 physical model

e) 3D scan of prototype 2

f) Updated curve network and deformed subdivision surface to represent prototype 2

FIGURE 11. Real mouse design example.

demonstrate how the existing iterative design process can be improved.

ACKNOWLEDGMENT

We thank Adriana Garcia for her help with the physical prototypes. This research is supported by National Science Foundation grant CAREER CMMI-0846730 and grant CMMI-1031703.

REFERENCES

[1] Shah, J. J., 2001. "Collaborative sketching (c-sketch) - an idea generation technique for engineering design". *Journal of Creative Behavior,* ***35***(3).

[2] Bove, L., 2012. The importance of modeling clay in transportation design. Available at http://luciano63.expertscolumn.com/article/importance-modeling-clay-transportation-design.

[3] 3dcoat software. Available at http://3d-coat.com.

[4] Topogun software. Available at http://www.topogun.com/.

[5] Loop, C., and Schaefer, S., 2008. "Approximating catmull-clark subdivision surfaces with bicubic patches". *ACM Transactions on Graphics (TOG),* ***27***(1), p. 8.

[6] Rusinkiewicz, S., and Levoy, M., 2001. "Efficient variants of the icp algorithm". In 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on, IEEE, pp. 145–152.

[7] Pottmann, H., Huang, Q.-X., Yang, Y.-L., and Hu, S.-M.,

2006. "Geometry and convergence analysis of algorithms for registration of 3d shapes". *International Journal of Computer Vision,* ***67***(3), pp. 277–296.

[8] Gelfand, N., Ikemoto, L., Rusinkiewicz, S., and Levoy, M., 2003. "Geometrically stable sampling for the icp algorithm". In 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on, IEEE, pp. 260–267.

[9] Brown, B. J., and Rusinkiewicz, S., 2007. "Global non-rigid alignment of 3-d scans". *ACM Transactions on Graphics (TOG),* ***26***(3), p. 21.

[10] Li, H., Sumner, R. W., and Pauly, M., 2008. "Global correspondence optimization for non-rigid registration of depth scans". In Computer graphics forum, Vol. 27, Wiley Online Library, pp. 1421–1430.

[11] Pauly, M., Mitra, N. J., Giesen, J., Gross, M. H., and Guibas, L. J., 2005. "Example-based 3d scan completion.". In Symposium on Geometry Processing, pp. 23–32.

[12] Tevs, A., Bokeloh, M., Wand, M., Schilling, A., and Seidel, H.-P., 2009. "Isometric registration of ambiguous and partial data". In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, pp. 1185–1192.

[13] Allen, B., Curless, B., and Popović, Z., 2003. "The space of human body shapes: reconstruction and parameterization from range scans". In ACM Transactions on Graphics (TOG), Vol. 22, ACM, pp. 587–594.

[14] Chui, H., and Rangarajan, A., 2003. "A new point matching algorithm for non-rigid registration". *Computer Vision and Image Understanding,* ***89***(2), pp. 114–141.

[15] Bechmann, D., 1994. "Space deformation models survey". *Computers & Graphics,* ***18***(4), pp. 571–586.

[16] Bronstein, A. M., Bronstein, M. M., Bruckstein, A. M., and Kimmel, R., 2008. "Analysis of two-dimensional non-rigid shapes". *International Journal of Computer Vision,* ***78***(1), pp. 67–88.

[17] Wand, M., Jenke, P., Huang, Q., Bokeloh, M., Guibas, L., and Schilling, A., 2007. "Reconstruction of deforming geometry from time-varying point clouds". In Symposium on Geometry processing, Citeseer, pp. 49–58.

[18] Eckstein, I., Pons, J.-P., Tong, Y., Kuo, C.-C., and Desbrun, M., 2007. "Generalized surface flows for mesh processing". In Proceedings of the fifth Eurographics symposium on Geometry processing, Eurographics Association, pp. 183–192.

[19] Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., and Seidel, H.-P., 2004. "Laplacian surface editing". *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing - SGP '04*, p. 175.

[20] Botsch, M., Pauly, M., Gross, M., and Kobbelt, L., 2006. "Primo: coupled prisms for intuitive surface modeling". In Symposium on Geometry Processing, Vol. 256, pp. 11–20.

[21] Sorkine, O., and Alexa, M., 2007. "As-rigid-as-possible surface modeling". In Symposium on Geometry processing, Citeseer, pp. 109–116.

[22] Sumner, R. W., Schmid, J., and Pauly, M., 2007. "Embedded deformation for shape manipulation". *ACM Transactions on Graphics (TOG),* ***26***(3), p. 80.

[23] Ohtake, Y., Belyaev, A., and Seidel, H.-P., 2004. "Ridge-valley lines on meshes via implicit surface fitting". In ACM Transactions on Graphics (TOG), Vol. 23, ACM, pp. 609–612.

[24] Gregorski, B. F., Hamann, B., and Joy, K. I., 2000. "Reconstruction of b-spline surfaces from scattered data points". In Computer Graphics International, 2000. Proceedings, IEEE, pp. 163–170.

[25] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W., 1992. *Surface reconstruction from unorganized points*, Vol. 26. ACM.

[26] LavouŐ, G., Dupont, F., and Baskurt, A., 2005. "Subdivision surface fitting for efficient compression and coding of 3D models". In SPIE Visual Communications and Image Processing (VCIP), pp. 1159–1170.

[27] Dammertz, H., and Keller, A., 2008. "The edge volume heuristic - robust triangle subdivision for improved bvh performance". In Interactive Ray Tracing, 2008. RT 2008. IEEE Symposium on, pp. 155–158.

[28] Zbrush software. Pixologic. Available at http://www.pixologic.com/zbrush/.

[29] Takayama, K., Panozzo, D., Sorkine-Hornung, A., and Sorkine-Hornung, O., 2013. "Sketch-based generation and editing of quad meshes". *ACM Trans. Graph.,* ***32***(4), July, pp. 97:1–97:8.

[30] Huang, Q.-X., Adams, B., Wicke, M., and Guibas, L. J., 2008. "Non-rigid registration under isometric deformations". In Computer Graphics Forum, Vol. 27, Wiley Online Library, pp. 1449–1457.

[31] Orbay, G., and Kara, L. B., 2011. "Sketch-based modeling of smooth surfaces using adaptive curve networks". In Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling, SBIM '11, ACM, pp. 71–78.

[32] Halstead, M., Kass, M., and DeRose, T., 1993. "Efficient, fair interpolation using catmull-clark surfaces". In Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '93, ACM, pp. 35–44.

[33] Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. L., 2001. "Topology matching for fully automatic similarity estimation of 3d shapes". In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01, ACM, pp. 203–212.

[34] Rubner, Y., Tomasi, C., and Guibas, L. J., 1998. "A metric for distributions with applications to image databases". In

Computer Vision, 1998. Sixth International Conference on, IEEE, pp. 59–66.

[35] Leordeanu, M., and Hebert, M., 2005. "A spectral technique for correspondence problems using pairwise constraints". In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, Vol. 2, IEEE, pp. 1482–1489.

[36] MacCracken, R., and Joy, K. I., 1996. "Free-form deformations with lattices of arbitrary topology". In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM, pp. 181–188.